

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ О.В. Коваль

(підпис)

(ініціали, прізвище)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2019р.

## Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення  
за спеціалізацією Інженерія програмного забезпечення розподілених систем  
на тему: Підсистема чисельного моделювання критичних режимів роботи  
контактних тепло-масообмінних апаратів

Виконав: студент 6 курсу, групи ТВ-381мп  
Касянчук Віталій Володимирович

\_\_\_\_\_

Науковий керівник к.т.н., доцент Кузьменко І.М.

\_\_\_\_\_

Рецензент \_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_  
(підпис)

Київ - 2019

**Національний технічний університет України  
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ

\_\_\_\_\_  
(прізвище, ініціали)      \_\_\_\_\_  
(підпис)  
« \_\_\_\_ » \_\_\_\_\_ 201 р.

ЗАТВЕРДЖУЮ

Завідувач кафедри

Коваль О.В.

\_\_\_\_\_  
(прізвище, ініціали)      \_\_\_\_\_  
(підпис)  
« \_\_\_\_ » \_\_\_\_\_ 2019р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Касянчук Віталій Володимирович

(прізвище, ім'я, по батькові)

1. Тема дисертації Підсистема чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів

Науковий керівник Кузьменко Ігор Миколайович, кандидат технічних наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “4” листопада 2019 року № 3813-с

2. Строк подання студентом дисертації “9” грудня 2019 року

3. Об'єкт дослідження – аналіз моделей двофазних течій

4. Предмет дослідження – системи аналізу моделей двофазних течій та їх види.

5. Перелік питань, які потрібно розробити:

- провести аналіз існуючих рішень;
- обрати бібліотеку для роботи з базою даних;
- провести аналіз Entity Framework, яка містить інструменти для роботи с бд на рівні сутностей;
- провести аналізу .Net Framework для побудови додатку за клієнт-серверною архітектурою;
- програмно реалізувати алгоритм однієї з моделей двофазних течій а саме на формулах кільцевих течії

---

6. Перелік ілюстративного матеріалу 1 додаток, 36 рисунків, 13 таблиць

7. Перелік публікацій

Касянчук В. В. Підсистема чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів

[Електронний ресурс] / В. В. Касянчук – 2019. – Режим доступу до ресурсу: <https://drive.google.com/file/d/RtRYpHeadLEchLotePuPEctATiANFeAutACIDiaM/view>.

8. Дата видачі завдання «12» вересня 2019 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Постановка цілей та завдань магістерської дисертації	12 вересня 2018 р. – 10 жовтня 2018 р.	
2	Аналітичний огляд літератури, бази джерел за темою дисертації	10 жовтня 2018 р. – 14 січня 2019 р.	
3	Вивчення питань з методології та методики написання магістерської дисертації	14 січня 2019 р. – 25 січня 2019 р.	
4	Підготовка концепції дисертації	25 січня 2019 р. – 20 лютого 2019 р.	
5	Написання основних розділів автореферату	20 лютого 2019 р. – 1 березня 2019 р.	
6	Підготовка основних розділів дисертації	26 квітня 2019 р. – 10 серпня 2019 р.	
7	Підготовка стартап-проекту	10 серпня 2019 р. – 2 вересня 2019 р.	
8	Переддипломна практика	2 вересня 2019 р. – 25 жовтня 2019 р.	
9	Захист програмного продукту	25 жовтня 2019 р.	
10	Участь у науковій конференції	14 листопада 2019 р.	
11	Попередній захист роботи	22 листопада 2019 р.	
12	Оформлення диплому	22 листопада 2019 р. – 9 грудня 2019 р.	
13	Здача всіх матеріалів на підпис зав кафедрою	9 грудня 2019 р.	
14	Захист магістерської дисертації	19 грудня 2019 р.	

Студент

\_\_\_\_\_

( підпис )

\_\_\_\_\_

(прізвище та ініціали)

Науковий керівник

\_\_\_\_\_

( підпис )

\_\_\_\_\_

(прізвище та ініціали)

## **РЕФЕРАТ**

### **на магістерську дисертацію**

**Структура й обсяг дипломної роботи.** Магістерська дисертація складається зі вступу, 6 розділів, висновку, переліку посилань з 12 найменувань, 1 додатку, і містить 26 рисунків, 13 таблиці. Повний обсяг магістерської дисертації складає 68 сторінок, з яких перелік посилань займає 2 сторінки, додатки – 4 сторінки.

**Актуальність теми.** У хімічній, нафтовій техніці і енергетиці застосовуються контактні теплообмінні апарати без поділяючої перегородки.

Завдяки відсутності перегородки вдається реалізувати масообмін або інтенсифікувати теплообмін. Однак, в таких апаратах протікають складні гідродинамічні процеси.

Зокрема, в залежності від витрат потоків в апараті розрізняють наступні гідродинамічні режими руху - плівковий, підвисання, захлинання (барботаж), винесення рідини. При цьому перехід до захлинання в плівкових апаратах розглядається як критичний режим, який обмежує їх роботу. Тому завдання

дослідити саме такий режим роботи та створити підсистему для розрахунку критичних режимів роботи є важливим та актуальним.

**Мета дослідження** полягає у підсистемі чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів.

**Завдання дослідження.** Для досягнення поставленої задачі були поставлені наступні завдання:

- провести аналіз існуючих рішень;
- обрати бібліотеку для роботи з базою даних;

- провести аналіз Entity Framework, яка містить інструменти для роботи с бд на рівні сутностей;
- провести аналізу .Net Framework для побудови додатку за клієнт-серверною архітектурою;
- програмно реалізувати алгоритм однієї з моделей двофазних течій а саме на формулах кільцевих течії

**Об’єктом дослідження** аналіз моделей двофазних течій.

**Предметом дослідження** системи аналізу моделей двофазних течій та їх види.

**Наукова новизна одержаних результатів.** Наукова новизна полягає у використанні у системі алгоритму базованому на формулах кільцевої течії.

**Практичне значення.** Підсистема чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів дозволяють виконати розрахунки результатів та перевірити алгоритм.

**Ключові слова:** *ПІДСИСТЕМА, ДВОФАЗНІ ТЕЧІЇ, ТЕПЛО-МАСООБМІННИЙ АПАРАТ, КІЛЬЦЕВА ТЕЧІЯ.*

# ABSTRACT

## for a master's dissertation

**Structure and volume of the thesis.** The master's thesis consists of an introduction, 6 sections, conclusion, a list of references of 12 names, 1 appendix, and contains 26 figures, 13 tables. The full volume of the master's thesis is 68 pages, of which the list of links occupies 2 pages, the annexes - 4 pages.

**Actuality of theme.** In chemical, petroleum engineering and energy, contact heat exchangers are used without a partition.

Due to the absence of a partition, it is possible to realize mass exchange or to intensify heat exchange. However, such devices have complex hydrodynamic processes.

In particular, depending on the flow rates in the apparatus distinguish the following hydrodynamic modes of motion - film, overhang, choke (bubbling), the removal of fluid. At the same time, the transition to choke in film machines is considered as a critical mode that limits their operation. Therefore, the task

to study such a mode of operation and to create a subsystem for the calculation of the critical modes of operation is important and relevant.

**The purpose of the study** is to provide subsystems of numerical simulation of critical modes of operation of contact heat-mass transfer devices.

**Objectives of the study.** To achieve this task, the following tasks were set:

- to analyze existing solutions;
- choose a library to work with the database;
- conduct an Entity Framework analysis that contains entity-level tools for working with DB;
- Conduct a .Net Framework analysis to build an application on a client-server architecture;

- programmatically implement the algorithm of one of the models of two-phase flows, namely, on formulas of circular flows

**The object of the study** is the analysis of models of two-phase flows.

**The subject of the study** is the system of analysis of models of two-phase currents and their types.

**Scientific novelty of the obtained results.** The scientific novelty is to use an algorithm based on circular flow formulas.

**Practical meaning.** The subsystem of numerical simulation of critical modes of operation of contact heat-mass transfer apparatus allows to perform calculations of results and to check the algorithm.

**Keywords:** *SUBSYSTEM, TWO-PHASE FLOWS, HEAT-MASS-EXCHANGE APPARATUS, RING FLOW.*

## ЗМІСТ

Вступ.....	9
1. Концепція клієнт-серверної архітектури .....	10
1.1. Сервер.....	10
1.2. Клієнт.....	13
1.3. Мережа .....	14
2. Опис предметної області .....	16
2.1. Двошвидкісний режим течії газорідинної суміші.....	20
3. Засоби розробки .....	30
3.1. Огляд .Net Framework .....	30
3.2. Використання середовища розробки Microsoft Visual Studio .....	31
3.3. Використання Entity Framework .....	31
3.4. Використання SQL Server .....	32
3.5. Використання NuGet .....	33
4. Опис програмної реалізації.....	36
4.1. Реалізація програмного продукту.....	36
5. Методика роботи користувача з програмною системою .....	48
5.1. Системні вимоги та запуск .....	48
6. Розроблення стартап проекту .....	53
6.1. Опис ідеї проекту .....	53
Висновки.....	65



## ВСТУП

У хімічній, нафтовій техніці і енергетиці застосовуються контактні теплообмінні апарати без розділяючої перегородки.

Завдяки відсутності перегородки вдається реалізувати масообмін або інтенсифікувати теплообмін. Однак, в таких апаратах протікають складні гідродинамічні процеси.

Зокрема, в залежності від витрат потоків в апараті розрізняють наступні гідродинамічні режими руху - плівковий, підвисання, захлинання (барботаж), винесення рідини. При цьому перехід до захлинання в плівкових апаратах розглядається як критичний режим, який обмежує їх роботу. Тому завдання

досліджити саме такий режим роботи та створити підсистему для розрахунку таких критичних режимів роботи так як критичний режим є важливим та актуальним.

Розв'язком перелічених проблем є використання мови програмування C# як мови програмування для серверної частини, оскільки на даний момент ця мова є досить зручною для таких видів задач і для неї існує багато перевірених часом рішень. Для побудови архітектури на серверній стороні було використано .Net-framework. Використання баз даних Sql Server

# 1. КОНЦЕПЦІЯ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- Сервер;
- Клієнт;
- Мережа.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів[1].

## 1.1. Сервер

Сервер — програма, що надає деякі послуги іншим програмам (клієнтам). Зв'язок між клієнтом і сервером зазвичай здійснюється за допомогою передачі повідомлень, часто через мережу, і використовує певний протокол для кодування запитів клієнта і відповідей сервера.[2] Серверні програми можуть бути встановлені як на серверному, так і на персональному комп'ютері, щоразу вони забезпечують виконання певних служб (наприклад, сервер баз даних чи веб-сервер).

Сервери - відповідають на запити за якими до них звертаються. Це може бути редагування інформації, додавання, видалення і т. д.

В залежності від запитів, існує багато протоколів для їх обробки. Наприклад при роботі з відображенням веб - сторінки разом з скриптами - протокол HTTP або HTTPS, або протокол передачі файлів з сервера на комп'ютер - протокол FTP.

Важливими факторами в роботи серверу є отримання термінової відповіді від серверу, кількість одночасних запитів і т. д.

Для того, щоб сервер міг працювати з даними, він підключається до бази даних, завдання якої - зберігання структурованої інформації і швидкий пошук інформації за критерієм. Сервери очікують запити від клієнтських програм і надають їм свої ресурси у вигляді даних (наприклад, завантаження файлів за допомогою HTTP, FTP )або ж у вигляді сервісних функцій (наприклад, робота з електронною поштою, спілкування за допомогою засобів миттєвого обміну повідомленнями або перегляд веб-сторінок).

Сервери баз даних використовуються для виконання користувацьких запитів до баз даних. При цьому СУБД (Система управління базою даних) знаходиться на сервері до якого підключаються клієнтські застосунки.

Сервери не завжди виконують одну роль а можуть виконувати декілька.

Основні типа серверів є такі:

- Remote Access/VPN Server

Сервери віддаленого доступу і VPN надають точку входу в вашу мережу для віддалених користувачів. Використовуючи роль Remote Access /

VPN Server, ви можете реалізувати протоколи маршрутизації для середовищ LAN і WAN. Ця роль підтримує модемні з'єднання і VPN через інтернет.

- Domain Controller

Контролер домену містить базу даних Active Directory. Контролери домену надають служби аутентифікації для користувачів і комп'ютерів, а також керують доступом до мережевих ресурсів. Роль контролера домену замінює інструмент DCPROMO, який був в Win2K. Ця роль дозволяє додати контролер домену до існуючого домену, створити новий домен, створити нове дерево.

- DNS Server

Служба DNS дозволяє перетворювати доменні імена (FQDN) в адреси IP. Версія DNS в WS2K3 включає службу динамічного DNS (DDNS), яка дозволяє комп'ютерам самим реєструватися в базі даних DNS.

- DHCP Server

Сервер DHCP дозволяє клієнтам отримувати свій IP за потребою. Сервер DHCP також надає додаткову інформацію для конфігурації мережі — адреса серверів DNS, WINS і т. ін.

- Streaming Media Server

Потоковий сервер надає служби Windows Media Services мережевим клієнтам. Windows Media Services використовуються для управління і доставки мультимедійного контенту — потокового відео та аудіо — через інтранет або інтернет

- WINS Server

WINS дозволяє клієнтам NetBIOS перетворювати імена комп'ютерів в адреси IP. На відміну від DNS, що вимагає доменні імена, WINS спроектована для внутрішньої інтрамережі для дозволу простих імен NetBIOS. Хоча можна мати мережу Windows без NetBIOS і WINS, багато утиліти все ще залежать від бази даних WINS. Багато типів записів, наявні в WINS, відсутні в DNS. Ці типи дозволяють легко знаходити в мережі сервери, які виконують специфічні служби (включаючи Terminal Services). Такий утилітою є Terminal Server Administration. Без WINS вам доведеться вручну вказувати сервер для управління.

## 1.2. Клієнт

Клієнт — апаратний або програмний компонент обчислювальної системи, який надсилає запити серверу.

Програма-клієнт взаємодіє з сервером, використовуючи певний протокол. Вона може запитувати з сервера будь-які дані, маніпулювати даними безпосередньо на сервері, запускати на сервері нові процеси і т. п. Отримані від сервера дані клієнтська програма може надавати користувачеві або використовувати як-небудь інакше, в залежності від призначення програми. Програма-клієнт і програма-сервер можуть працювати як на одному і тому ж комп'ютері, так і на різних. У другому випадку для обміну інформацією між ними використовується мережеве з'єднання[3].

Різновидом клієнтів є термінали — робочі місця на багатокористувацьких ЕОМ, обладнані монітором та клавіатурою, і не здатні працювати без сервера. У 1990-ті роки з'явилися мережеві комп'ютери — щось середнє між терміналом і персональним комп'ютером. Мережеві

комп'ютери мають спрощену структуру і багато в чому залежать від сервера. Іноді терміналом називають будь-який клієнт, або тільки тонкий клієнт.

Тим не менш, не завжди під клієнтом мається на увазі комп'ютер зі слабкими обчислювальними ресурсами. Найчастіше поняття «клієнт» і «сервер» описують розподіл ролей при виконанні конкретного завдання, а не обчислювальні потужності. На одному і тому ж комп'ютері можуть одночасно працювати програми, що виконують як клієнтські, так і серверні функції. Наприклад, веб-сервер може як клієнт отримувати дані для формування сторінок від SQL-сервера.

### **1.3. Мережа**

Мережа - сукупність комп'ютерів, об'єднаних за допомогою каналів зв'язку та засобів комутації в єдину систему для обміну повідомленнями і доступу користувачів до програмних, технічних, інформаційних і організаційних ресурсів мережі.

Мережева модель OSI — абстрактна мережева модель для комунікацій і розробки мережевих протоколів.

Будь-який протокол моделі OSI повинен взаємодіяти або з протоколами свого рівня, або з протоколами на одиницю вище або нижче за свій рівень. Взаємодії з протоколами свого рівня називаються горизонтальними, а з рівнями на одиницю вище або нижче — вертикальними. Будь-який протокол моделі OSI може виконувати лише функції свого рівня і не може виконувати функцій іншого рівня, що не виконується в протоколах альтернативних моделей[4].

## **Висновок до розділу 1**

Розробка системи, що має клієнт-серверну архітектуру — один із найпопулярніших напрямів реалізації розподілених систем, причому створюються вони не лише для специфічних цілей. Можна зробити висновок, що клієнт-серверна архітектура стала невід’ємною частиною розподілених систем, сфери її застосування постійно збільшуються, що, в свою чергу, вимагає більш детального вивчення питань, пов’язаних з забезпеченням якісної взаємодії між всіма компонентами розподілених систем.

## 2. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Фазою називають один зі станів речовини, яка може бути газоподібним, рідким або твердим. Багатофазна течія - це спільна течія декількох фаз. Двофазний потік являє собою окремий випадок багатофазного течії, що включає в себе два різнорідних компонента. Це рідини з твердими або газовими включеннями, гази з краплями рідини або твердими частинами. У природі багатофазними середовищами є дим, туман, смог, дощ і т.д. У техніці типовим прикладом двофазного потоку є потік пара з краплями рідини. Навіть при адіабатичному плинні такої суміші у вертикальному каналі краплі рідини стикаються зі стінкою, утворюючи стікаючу плівку рідини. Стікаюча плівка взаємодіє з зустрічним потоком пара[5].

Поява плівки змінює гідродинаміку течії. Так, при польоті літака в зоні дощу плівка рідини на поверхні викликає зміни в прикордонному шарі, провокуючи передчасний відрив потоку з несучих поверхонь і збільшує лобовий опір. Подібні течії зустрічаються в каналах при газифікації криогенного палива, в енергетиці та хімічній промисловості при використанні пару. Тому, вивчення таких явищ є актуальним завданням.

Найважливішими характеристиками в течіях двофазних сумішей є масові та об'ємні частки фаз відповідно в масовому і об'ємній витраті суміші. Введемо поняття об'ємної концентрації легкої фази - це частка обсягу суміші, яку займає газом.

Розглянемо поперечний переріз каналу, тоді середня, об'ємна концентрація (об'ємний газовміст) легкої фази буде визначатися за формулою

$$\varphi = \frac{S_1}{S} \quad (2.1)$$



Де  $S_1$  - площа поперечного перерізу, яку займає газ,  $S$  - площа поперечного перерізу каналу. Значення необхідно для визначення ваги вертикального стовпа двофазної рідини для аналізу силового взаємодії потоку зі стінкою каналу. Середня концентрація рідкої фази - це частка обсягу суміші, яку займає рідина. Вона буде виражатися величиною  $(1 - \varphi)$ . Щільність суміші запишеться у вигляді.

$$\rho_{см} = \rho_1 \varphi + \rho_2 (1 - \varphi) \quad (2.2)$$

Де індексами «1» і «2» будемо позначати відповідно газ і краплі рідини, індекс «см» позначає суміш,  $\rho$  - щільність фази.

Об'ємний газовміст відрізняється від об'ємного витратного газовмісту через відносний рух фаз. Видаткова, об'ємна концентрація (газовміст) визначається виразом

$$\beta = \frac{Q_1}{Q_1 + Q_2} \quad (2.3)$$

де  $Q$ - об'ємна витрата фази.

Видаткова масова концентрація пов'язана з видатковою об'ємною концентрацією через щільності фаз

$$X = \frac{Q_1 \rho_1}{Q_1 \rho_1 + Q_2 \rho_2} = \frac{G_1}{G_1 + G_2} \quad (2.4)$$

Де  $G$  - масова витрата фази.

Використовується поняття масової концентрації

$$C_1 = \frac{\rho_1 \varphi}{\rho_1 \varphi + \rho_2 (1 - \varphi)} \quad C_2 = \frac{\rho_2 (1 - \varphi)}{\rho_1 \varphi + \rho_2 (1 - \varphi)} \quad (2.5)$$

Динамічна в'язкість суміші обчислюється таким чином

$$\mu_{см} = \mu_1 X + \mu_2 (1 - X) \quad (2.6)$$

Де  $\mu$  - в'язкість фаз.

Якщо розміри крапель прямо пропорційні з розмірами молекул, то потік можна вважати однорідним а швидкості газу та крапель рідини в такому потоці однаковими. Таку модель течії називають квазігомогенною і говорять

про двошвидкісний режим течії, маючи на увазі швидкість суміші і швидкість рідкої плівки. Для квазігомогенну потоку об'ємна концентрація збігається з видаткової об'ємною концентрацією  $\varphi = \beta$ . У разі чималих крапель середовище вже не є однорідною і розглядається тришвидкісна модель дисперсно-плівкової течії. Такий перебіг має три характерні швидкості: швидкість газу, швидкість крапель і швидкість рідини в плівці. Зауважимо, що швидкість крапель завжди менше швидкості газу.

Для неоднорідного середовища вводиться поняття величини дрейфу або швидкості ковзання фаз

$$y_1 = \bar{V}_1 - \bar{V}_{\text{см}} \quad y_2 = \bar{V}_2 - \bar{V}_{\text{см}} \quad (2.6)$$

Де лінія зверху означає середню величину.

Двофазний потік характеризується приведеними швидкостями або, тою ж самою, щільністю об'ємної витрати

$$\bar{V}_{10} = \bar{V}_1 \varphi \quad \bar{V}_{20} = \bar{V}_2 (1 - \varphi) \quad (2.7)$$

З точки зору гідродинаміки - це такі швидкості, які мали б фази при збереженні витрати і повному заповненні перетину каналу. Тоді швидкість суміші запишеться у вигляді

$$\bar{V}_{\text{см}} = \bar{V}_{10} \varphi + \bar{V}_{20} = \bar{V}_1 \varphi + \bar{V}_2 (1 - \varphi) \quad (2.8)$$

Двофазні течії підкоряються всім основним законам гідромеханіки, однак рівняння більш складні і численні. Крім того, рівняння руху для двофазних течій є незамкненими, і важливим завданням при математичному моделюванні є вибір додаткових співвідношень. Аналізуючи існуючі методи розрахунку двофазних потоків, можна відзначити, що вони спираються на великий експериментальний матеріал. Це й не дивно, з огляду на різноманіття режимів течії двофазної суміші.

Зустрічаються на практиці режими течії газорідних сумішей в каналах можна поділити на види в залежності від об'ємної концентрації

легкої фази. По мірі збільшення об'ємної концентрації газу це будуть: бульбашковий, снарядний, дисперсно-плівковий і дисперсний режими.

Бульбашковий режим існує при об'ємної концентрації газу  $\varphi > 0,2 - 0,3$ . При великих концентраціях відбувається злиття бульбашок з утворенням пухирів снарядоподібної форми, які займають майже весь поперечний переріз каналу. Таким чином, бульбашковий режим переходить в снарядний. Якщо швидкість газової фази досить велика  $\bar{V}_1 > 5 - 10$  м/с, то снарядна структура потоку стає нестійкою; на стінці каналу з'являється плівка рідини, а ядро потоку має піноподібну структуру.

При подальшому збільшенні об'ємної газової концентрації  $\varphi > 0,6 - 0,8$  реалізується плівковий або кільцевої режим течії, при якому рідка фаза утворює безперервну плівку, що тече по стінці каналу, а газова фаза - ядро потоку. Через динамічні взаємодії газового ядра потоку і рідкої плівки на поверхні останньої утворюються хвилі, з гребенів яких можуть зриватися краплі і нестися в ядро потоку. В цьому випадку реалізується дисперсно-плівковий або дисперсно-кільцевий режим течії. При підведенні тепла із зовнішнього середовища плівка випаровується і дисперсно-кільцевий режим течії переходить в просто дисперсний - протягом суміші пару і крапель.

При течії газорідинної суміші в вертикальних каналах для всіх режимів цієї течії має місце практично вісесиметричний розподіл концентрацій і швидкостей фаз по перетину. При перебігу в горизонтальних і похилих каналах через гравітацію порушується осьова симетрія у розподілі фаз по перетину.

Побудова математичної моделі кільцевого течії, навіть для усереднених характеристик, представляє значні труднощі, оскільки завдання формулюється як спряжена, що враховує взаємодію двухфазного ядра потоку з рідкої плівкою. Кількість невідомих величин в цьому випадку доходить до десяти: це середні швидкості фаз, середня швидкість рідини в плівці, швидкість на поверхні плівки, міжфазне тертя і тертя на стінці, радіус і

товщина плівки, втрати тиску по довжині каналу, середні об'ємна, видаткова та масова концентрації. Тому, використання диференціальних співвідношень, що описують двофазні потоки, не завжди виправдано і призводить до суттєвого ускладнення завдання.

Для деяких, практично важливих, завдань досить визначити інтегральні характеристики течії, такі як середні швидкості, середні концентрації в поперечному перерізі каналу, втрати тиску по довжині і тертя. Поставимо задачу математичного моделювання двофазних течій, для яких досить визначення інтегральних характеристик. Тоді основною проблемою буде складання замкнутої системи рівнянь, що описують ту чи іншу течію. Оскільки завдання вирішується чисельно, то виникає проблема вибору початкового наближення, що саме по собі викликає певні труднощі. Зазвичай початкові наближення задають з фізичних міркувань, а для збіжності рішення використовують метод послідовних наближень.

## **2.1. Двошвидкісний режим течії газорідинної суміші**

Режим течії пароводяної суміші називається двошвидкісним, якщо по стінці каналу стікає плівка рідини, а ядро потоку являє собою квазіоднорідну суміш, що складається з пару та дрібних крапель рідини. В цьому випадку можна скористатися теорією гомогенної течії, яка дає найпростіший метод дослідження двофазних потоків. Пароводяна суміш розглядається, як квазіконтінуум, для якої визначаються середні характеристики течії.

Розглянемо висхідну течію пароводяної суміші у вертикальному циліндричному каналі. Потік вважатимемо квазігомогенним, а процес адіабатичним. Ядро потоку може являти собою сухий або вологий пар (з краплями рідини). Плівка має постійну товщину і гладку поверхню розділу.

Тоді для сталого процесу характеристики течії будуть однакові в будь-якому поперечному перерізі каналу

Нехай витрати вологої пари в каналі і рідини в плівці пов'язані співвідношенням

$$G = \pi R_1^2 \bar{V}_{\text{см}} p_{\text{см}} + \pi (R^2 - R_1^2) \bar{V}_3 p_3 \quad (2.9)$$

де  $R_1$ - радіус плівки, індексом «3» будемо позначати параметри рідини в плівці.

Якщо рідина в плівку надходить ззовні, тоді її витрата не залежить від витрати пароводяної суміші і визначається формулою

$$G_3 = \pi(R^2 - R_1^2) \bar{V}_3 \rho_3 \quad (2.10)$$

При висхідному потоку суміші збільшення швидкості потоку призводить до того, що поверхневі шари рідини в плівці спочатку загальмовуються, а потім прагнуть вгору зустрічним потоком.

На рисунку 2.1 зображено ділянку вертикального адіабатичного каналу, по внутрішній стінці якого стікає плівка рідини, що взаємодіє із зустрічним потоком суміші. Показаний момент формування плівки: початкова фаза перебігу - коли плівка ще не встигла сформуватися; сформована течія і проміжний режим несформованої течії.

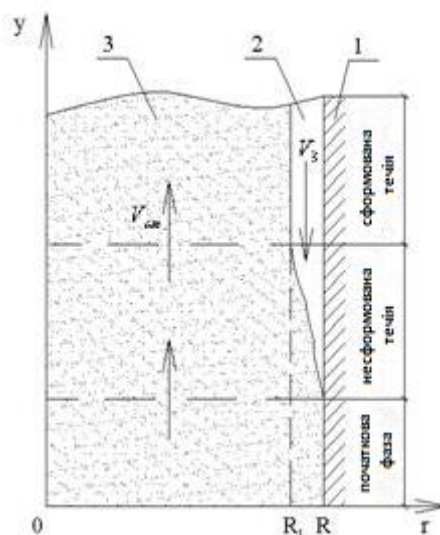


Рисунок 2.1 — Схема висхідного течії двофазної суміші

Де 1 - стінка, 2 - плівка рідини, 3 - дисперсний потік,  $v$  - швидкість рідини в плівці,  $\bar{V}_3$  - швидкість суміші.

Взаємодія плівки рідини із зустрічним потоком здійснюється через напруга тертя на поверхні розділу фаз. Виділимо зовнішню задачу про перебіг пароводяної суміші і внутрішню задачу про перебіг плівки по стінці каналу, а їх поєднання виконаємо через швидкість на поверхні плівки і напруга тертя на поверхні розділу фаз. Епюри швидкостей для протилежних течій суміші і рідкої плівки представлені на рисунку 2.2

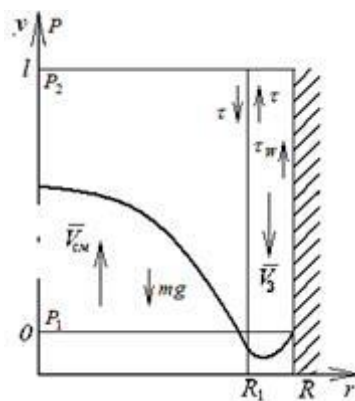


Рисунок 2.2 — Зустрічні потоки

Де  $\tau, \tau_w$  - напруги тертя на межі поділу фаз і на стінці,  $\bar{V}_{cm}, \bar{V}_3$  - середні швидкості суміші і плівки рідини,  $l$  - довжина ділянки каналу,  $P_1, P_2$  - тиску в двох перетинах

Сформулюємо математичну модель сформованої висхідної течії пароводяної суміші, що контактує з плівкою рідини. Рідина в плівку подається примусово, тобто витрати суміші і рідини не пов'язані. Для ідентифікації завдання задамо масові витрати рідини в плівці  $G_3$ , радіус каналу  $R$  та температуру процесу для визначення теплофізичних характеристик.

Суміш розглядаємо, як однорідну рідину; Течія – ламінарна. Тоді математичне формулювання зовнішнього завдання має вигляд (вісь  $y$  спрямована вгору за течією суміші)

$$\frac{v_{\text{см}}}{r} \frac{d}{dr} \left( r \frac{dV_{\text{см}}}{dr} \right) = g + \frac{1}{p_{\text{см}}} \frac{dP_{\text{см}}}{dy}$$

$$r = 0, \frac{dV_{\text{см}}}{dr} = 0, r = R_1, V_{\text{см}} = V_{31} \quad (2.11)$$

Граничні умови задачі відповідають максимальній швидкості на осі каналу і рівності швидкостей на межі розділу фаз.

Вирішуючи крайову задачу, знайдемо напруження тертя на межі поділу фаз, яке, в подальшому, перенесемо на поверхню плівки

$$t = \frac{4u_{\text{см}}(\bar{V}_{\text{см}} \pm V_{31})}{R_1} \quad (2.12)$$

Де  $V_{31}$ - швидкість суміші на межі розділу фаз (тут і далі в формулах для відносної швидкості слід брати «+», якщо швидкості  $\bar{V}_{\text{см}}$  і  $V_{31}$  різноспрямовані, і «-», якщо швидкості  $\bar{V}_{\text{см}}$  і  $V_{31}$  однонаправлені).

$v_{\text{см}} = u_{\text{см}}/p_{\text{см}}$  – динамічна в'язкість,  $g$ - прискорення вільного падіння

Поставимо задачу про ламінарну течію рідкої плівки по стінці вертикального циліндричного каналу під дією сили тяжіння і сил тертя (вісь  $y$  спрямована вниз). Рівняння руху має такий вигляд

$$0 = g - \frac{1}{p_3} \frac{dP_3}{dy} + \frac{v_3}{r} \frac{d}{dr} \left( r \frac{dV_3}{dr} \right) \quad (2.13)$$

Гradient тиску  $\frac{dP_3}{dy}$  висловимо з умови рівності тиску в поперечному перерізі

каналу для суміші і рідкої плівки тобто  $\frac{dP_3}{dy} = \frac{dP_{\text{см}}}{dy}$  де  $P_{\text{см}}$ - тиск в суміші.

Виділимо обсяг каналу, обмежений двома поперечними перетинами, відстань між якими  $l$  та розглянемо сили, що діють на виділений обсяг суміші і плівки.

На суміш в напрямку руху діє перепад тиску; в зворотному напрямку діє сила тяжіння і поверхнева сила тертя. Тоді баланс сил запишеться в наступному вигляді



$$(P'_{\text{см}} - P''_{\text{см}})\pi R_1^2 = \pi R_1^2 l p_{\text{см}} g + 2\pi R_1 l t$$

або

$$\frac{dP_{\text{см}}}{dy} = \frac{P'_{\text{см}} - P''_{\text{см}}}{l} = \frac{1}{R_1} (R_1 p_{\text{см}} g + 2t) \quad (2.14)$$

де  $P'_{\text{см}}$  і  $P''_{\text{см}}$  - тиск в виділених поперечних перетинах каналу. Для ламінарного потоку сила тяжіння значно перевершує тертя на міжфазній поверхні, тому отримаємо наближений вираз  $\frac{dP_{\text{см}}}{dy} = \frac{dP_3}{dy} = P_{\text{см}} g$  і, тоді математична формулювання завдання для стікаючої плівки запишеться у вигляді

$$\frac{u_3}{r} \frac{d}{dr} \left( r \frac{dV_3}{dr} \right) = g(p_{\text{см}} - p_3)$$

$$r = R_1, t = u_3 \frac{dV_3}{dr}, r = R, V_3 = 0 \quad (2.15)$$

де граничні умови визначають тертя  $t$  на поверхні плівки, отримане з рішення зовнішньої завдання, і умова прилипання рідини на стінці каналу.

Вирішення крайової задачі (2.15) дає швидкість рідини в плівці

$$V_3(r) = \frac{g\varphi(p_3 - p_1)}{4u_3} (R^2 - r^2) + \frac{R_1}{u_3} \left( \frac{gR_1}{2} \varphi(p_3 - p_1) + t \right) \ln \frac{r}{R} \quad (2.17)$$

Тоді середня швидкість рідини буде визначатись виразом

$$\overline{V}_3 = \frac{2}{R^2 - R_1^2} \int_{R_1}^R V(r) r dr \quad (2.18)$$

а швидкість рідини на поверхні плівки запишеться у вигляді

$$V_{31} = \frac{g\varphi(p_3 - p_1)}{4u_3} (R^2 - R_1^2) + \frac{R_1}{u_3} \left( \frac{gR_1}{2} \varphi(p_1 - p_3) + t \right) \ln \frac{R_1}{R} \quad (2.19)$$

Тертя на стінці каналу знайдемо відповідно до закону Ньютона  $t = u \frac{dV}{dr}$

$$t_w = \frac{g\varphi(p_3 - p_1)}{2} - \left( \frac{R_1}{R} \frac{g\varphi R_1}{2} (p_1 - p_3) + t \right) \quad (2.20)$$

Тоді, математична модель кільцевої течії включатиме в себе систему рівнянь (2.10), (2.12), (2.18) - (2.20).

Нехай течія суміші має турбулентний характер. Тоді для вирішення зовнішньої завдання скористаємося законом Блазіуса для міжфазного тертя. Рухливу межу потоку врахуємо, використовуючи відносну швидкість на міжфазній поверхні, щільність суміші обчислимо за формулою (2)

$$t = \frac{0,3164}{\sqrt[4]{Re}} \frac{p_{cm} V_{від}^2}{8} \quad (2.21)$$

де  $Re = \frac{\bar{V}_{cm} 2R_1 p_{cm}}{u_1}$  - число Рейнольдса, обчислене по видатковій швидкості, - відносна швидкість.  $V_{від} = \bar{V}_{cm} \pm V_{31}$  - відносна швидкість

Запишемо баланс сил, що діють на виділений обсяг плівки (рис.2.2). На плівку в напрямку руху діє сила тяжіння; в зворотному напрямку діє перепад тиску і поверхневі сили тертя на міжфазній поверхні і стінці каналу. Тоді матимемо

$$(P'_3 - P''_3)\pi(R^2 - R_1^2) + 2\pi R_1 l t + 2\pi R_1 l t_w = \pi((R^2 - R_1^2)lp_3 g$$

$$\text{Або} \quad \frac{dP}{dy} = \frac{P'_3 - P''_3}{l} = p_3 g - \frac{2(R_1 t + R t_w)}{R^2 - R_1^2} \quad (2.22)$$

Використовуючи умову сталості тиску в поперечному перерізі каналу, прирівняємо праві частини виразів (2.14) і (2.21). Тоді, після деяких перетворень, отримаємо балансове рівняння для сил, що діють на виділені обсяги суміші і плівки.

$$\frac{(p_3 - p_1)g\varphi}{2} = \frac{R(R t \pm R_1 t_w)}{R_1(R^2 - R_1^2)} \quad (2.23)$$

де  $t_w$ - тертя на стінці каналу, у формулі (2.23) слід брати «+», якщо швидкості  $\bar{V}_{cm}$  і  $V_{31}$  різноспрямовані, і «-», якщо швидкості  $\bar{V}_{cm}$  і  $V_{31}$  однонаправлені.

Далі математична модель кільцевої (турбулентно-ламінальної) течії включатиме в собі систему нелінійних алгебраїчних рівнянь (2.10), (2.18), (2.19), (2.21) і (2.23).

Використовуючи вказану модель кільцевої течії, був виконаний розрахунок параметрів течії при наступних вхідних даних: висхідна двофазна суміш складається з пару та води при температурі  $200^\circ\text{C}$ ,  $p_1 = 7,87 \text{ кг/м}^3$ ,  $p_2 = 865 \text{ кг/м}^3$ ,  $p_3 = 865 \text{ кг/м}^3$ ,  $u_1 = 1,65 \times 10^{-5} \text{ Пас}$ ,  $u_2 = 0,14 \times 10^{-3} \text{ Пас}$ ,  $u_3 = 0,14 \times 10^{-3} \text{ Пас}$ , радіус каналу  $R = 10 \text{ мм}$ , концентрація легкої фази  $\varphi = 0,95$ , рідина в плівку подається примусово с витратою  $G_3 = 10^{-3} \text{ кг/с}$ . Розрахунки за вхідними даними наведені в таблиці 2.1

Таблиця 2.1

№	$\bar{V}_{cm},$ м/с	$\bar{V}_3,$ м/с	$V_{31},$ м/с	$R_1 \cdot 10^3$ м	$\tau,$ н/м <sup>2</sup>	$\tau_w,$ н/м <sup>2</sup>	$Re \times$ $\times 10^{-4}$	$Re_3$
1	1.4	0.153	0.168	9.879	0.289	-0.675	0.85	114
2	1.8	0.140	0.115	9.868	0.405	-0.647	1.09	114
3	2.0	0.134	0.085	9.861	0.467	-0.637	1.21	114
4	2.2	0.127	0.052	9.854	0.533	-0.630	1.33	114
5	2.4	0.121	0.017	9.846	0.601	-0.627	1.45	114
6	4.0	0.078	-0.38	9.762	1,184	-0.620	2.40	114

де  $Re_3 = \bar{V}_3(R - R_1)/\nu_3$  (течія в плівці ламинарна, якщо  $Re_3 \leq 400$ ).

З таблиці виходить, що зі збільшенням швидкості зустрічного потоку швидкість рідини в плівці, а також швидкість на поверхні плівки

зменшується. Тертя на міжфазній поверхні і товщин плівки збільшується. Числа Рейнольдса показують, що потік турбулентний, а плівка ламінарна

На рисунках 2.3 та 2.4 показані профілі швидкостей для режимів течій №1 та №3.

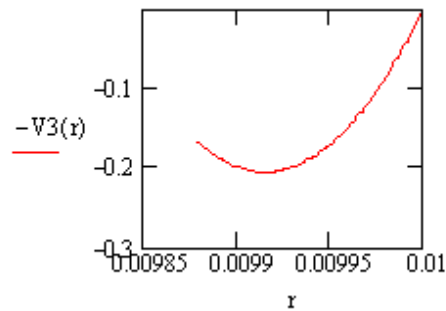


Рисунок 2.3 — Профіль швидкостей в плівці  
при впливі зустрічного потоку  
пару для  $Re = 8,5 \cdot 10^3$

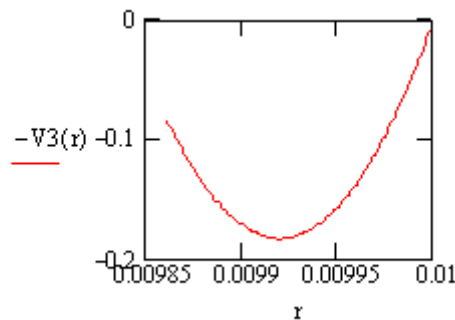


Рисунок 2.4 — Профіль швидкостей в плівці  
при впливі зустрічного потоку  
пару для  $Re = 10,9 \cdot 10^3$

З графіків видно, як зовнішній потік двофазної суміші впливає на стікаючу плівку: шари рідини на міжфазній поверхні сповільнюються, і максимум швидкості зсувається вглиб плівки

Якщо швидкість зустрічного потоку продовжує зростати, то швидкість рідини на поверхні плівки спочатку зменшиться до нуля, а потім на поверхні плівки виникає зворотна течія (режим №6) рисунок 2.5

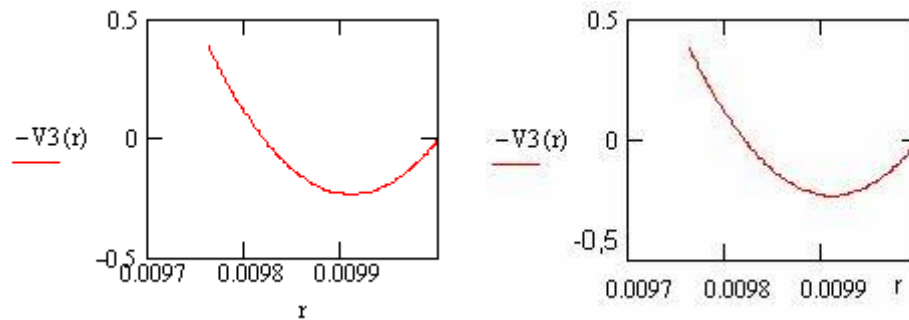


Рисунок 2.5 Розворот плівки під дією зустрічного турбулентного потоку пару  $Re = 24 \cdot 10^3$

У випадку, коли рідина в плівку надходить зовні, зворотня течія на поверхні плівки призводить до того, що кількість рідини в районі джерела накопичується и, в кінцевому результаті заповнює весь канал – такий режим призводить до захлинання течії[6].

## Висновок до розділу 2

У даному розділі було описано декілька режимів роботи контактних теплообмінних апаратів. Були розглянуті фази, міжфазні течії та їх види. Також було розглянуто одну з моделей двофазних течій. Було проведений аналіз які фактори можуть привести до захлинання течії

### 3. ЗАСОБИ РОЗРОБКИ

Під час розробки програмного продукту використовувалися .Net Framework, середовище розробки Microsoft Visual Studio, інструмент для роботи с БД Entity Framework та са ма бд SQL Server, та систему управління пакетами для .Net Framework — NuGet.

#### 3.1. Огляд .Net Framework

.Net Framework — програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків.

Середовище розробки .NET створює байт-код, призначений для виконання віртуальною машиною. Вхідна мова цієї машини в .NET називається CIL (Common Intermediate Language) також відома як MSIL (Microsoft Intermediate Language), або просто IL. Застосування байт-коду дозволяє отримати крос-платформність на рівні скомпільованого проекту (в термінах .NET: збірка), а не на рівні сирцевого тексту, як, наприклад, в C. Перед запуском збірки в середовищі виконання (CLR) байт-код перетворюється вбудованим в середовище JIT-компілятором (just in time, компіляція на льоту) в машинні коди цільового процесора.

C# — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтанутом та Пітером Гольде під егідою Microsoft Research (при фірмі Microsoft).

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на

функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато що від своїх попередників — мов C++, Delphi, Модула і Smalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++) [7].

### **3.2. Використання середовища розробки Microsoft Visual Studio**

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. [8]

### **3.3. Використання Entity Framework**

Entity Framework являє спеціальну об'єктно-орієнтовану технологію на базі фреймворка .NET для роботи з даними. Якщо традиційні засоби ADO.NET дозволяють створювати підключення, команди та інші об'єкти для взаємодії з базами даних, то Entity Framework являє собою більш високий рівень абстракції, який дозволяє абстрагуватися від самої бази даних і працювати з даними незалежно від типу сховища. Якщо на фізичному рівні ми оперуємо таблицями, індексами, первинними і зовнішніми ключами, то на

концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами.

Перша версія Entity Framework - 1.0 вийшла ще в 2008 році і представляла дуже обмежену функціональність, базову підтримку ORM (object-relational mapping - відображення даних на реальні об'єкти) і один єдиний підхід до взаємодії з бд - Database First. З виходом версії 4.0 у 2010 році багато чого змінилося - з цього часу Entity Framework став рекомендованою технологією для доступу до даних, а в сам фреймворк були введені нові можливості взаємодії з бд - підходи Model First і Code First.

Центральною концепцією Entity Framework є поняття сутності або entity. Сутність представляє набір даних, асоційованих з певним об'єктом. Тому дана технологія передбачає роботу не з таблицями, а з об'єктами і їх наборами. Будь-яка сутність, як і будь-який об'єкт з реального світу, має низку властивостей. Наприклад, якщо сутність описує людини, то ми можемо виділити такі властивості, як ім'я, прізвище, зріст, вік, вага. Властивості необов'язково представляють прості дані типу int, а й можуть представляти більш комплексні структури даних. І у кожної сутності може бути одна або кілька властивостей, які будуть відрізняти цю сутність від інших і будуть унікально визначати цю сутність. Подібні властивості називають ключами.[9]

### **3.4. Використання SQL Server**

Microsoft SQL Server — комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.



SQL Server - включає низку вдосконалень для роботи з критичними бізнес-застосунками і бізнес-аналітикою як в традиційних дата-центрах, так і в приватних, публічних і гібридних хмарних середовищах. Серед нових можливостей SQL Server 2012 виділяються SQL Server AlwaysOn (рішення підтримки високого рівня доступності даних та аварійного відновлення), xVelocity (технологія збільшення продуктивності сховищ даних та програм бізнес-аналітики), нові рішення в області візуалізації PowerPivot і PowerView для створення звітів і аналітичних програм з Excel і SharePoint, покращені інструменти для інтеграції даних і управління ними, включаючи SQL Server Data Quality Services і Master Data Services, нова семантична модель бізнес-аналітики та інструмент для адміністраторів баз даних і розробників застосунків SQL Server Data Tools[10].

Microsoft та інші компанії пропонують велику кількість програмних засобів розробки, які дозволяють розробляти застосунки для бізнесу з використанням баз даних Microsoft SQL Server. Microsoft SQL Server 2005 включає також Common Language Runtime (CLR) Microsoft .NET, що дозволяє застосункам, розробленим на мовах платформи .NET (наприклад, VB.NET або C#), реалізовувати процедури, що зберігаються та різні функції.

### **3.5. Використання NuGet**

NuGet — це вільна система керування пакунками, розроблена для Microsoft development platform (що раніше називалась NuPack). З моменту запуску у 2010-му, NuGet було включено до широкої системи інструментів та сервісів.

NuGet спочатку було презентовано як додаток для Visual Studio. У 2012 році NuGet був інстальований в Visual Studio за замовчуванням. NuGet також інтегровано з SharpDevelop. NuGet можна використовувати як з командної стрічки, так і автоматизовано, за допомогою скриптів.

Він підтримує багато з мов програмування, таких як:

- Пакети для .NET Framework;
- Нативні пакети, написані мовою C++, з можливістю створення пакетів на CoApp або OneGet.

Ключовий інструмент для будь-якої сучасної платформи розробки - це механізм, за допомогою якого розробники можуть створювати, передавати один одному і використовувати корисний код. Часто такий код розподілений по "пакетах", що включає скомпільований код (у вигляді DLL бібліотек) і інший вміст, необхідний проектам які використовують ці пакети.

Для .NET (в тому числі .NET Core) механізмом спільного використання коду, підтримуваним Майкрософт, є NuGet, який визначає, як створюються, розміщуються і використовуються пакети для .NET, а також надає інструменти для кожної з цих ролей.

Простіше кажучи, пакет NuGet є окремим ZIP-файл з розширенням .nupkg, який містить скомпільований код (DLL), інші файли, пов'язані з цим кодом, і описовий маніфест, що включає такі відомості, як номер версії пакета. Розробники, у яких є код, до якого потрібно надати загальний доступ, створюють пакети і публікують їх на закритих або відкритих вузлах. Споживачі отримують ці пакети з відповідних вузлів, додають їх у свої проекти, а потім викликають функції пакета в коді свого проекту. При цьому NuGet сам обробляє всі проміжні дані.

Так як NuGet підтримує закриті вузли поряд з відкритим вузлом [nuget.org](https://nuget.org), за допомогою пакетів NuGet ви можете ділитися кодом, використовуваним в рамках організації або робочої групи. Пакети NuGet також є зручним способом факторизувати свій код для використання тільки у власних проектах. Іншими словами, пакет NuGet є спільно використовуваної одиницею коду, проте не вимагає і не має на увазі будь-якого певного способу надання загального доступу[11].

### **Висновок до розділу 3**

Розробка системи з клієнт-серверною архітектурою — досить не тривіальна задача. Проте існує достатня кількість різноманітних інструментів, що роблять процес розробки легшим і зрозумілішим.

У данному розділі було розглянуто основні технології які були використані для створення системи. А саме:

.Net framework — програмна технологія яка ідеально підходить для розробки подібних систем. Середовище розробки .NET створює байт-код, призначений для виконання віртуальною машиною.

Visual Studio чудово підходить як середовище розробки для .Net framework та мови програмування C#.

Entity Framework який чудово справляється з роботою сутностей так і з таблицями в бд.

Можна зробити висновок, що клієнт-серверна архітектура стала невід’ємною частиною розподілених систем, сфери її застосування постійно збільшуються, що, в свою чергу, стимулює створення інструментів для полегшення розробки таких систем.

## 4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

На основі представленої у попередніх розділах інформації видно, що з метою інтерактивної реалізації клієнт-серверної архітектури необхідно реалізувати певні компоненти. Для написання підсистеми використовувалась мова програмування C#[12], Entity Framework для взаємодії підсистеми з базою даних та SQL Server[10].

Взагалі підсистему чисельного моделювання критичних режимів роботи тепло-масообмінних апаратів можна поділити на три частини. Це авторизація користувача, або його реєстрація в системі, власне розрахунки які виконує користувач, та робота з сервером бази даних.

### 4.1. Реалізація програмного продукту

Спочатку розглянемо діаграму прецедентів на рис 4.1

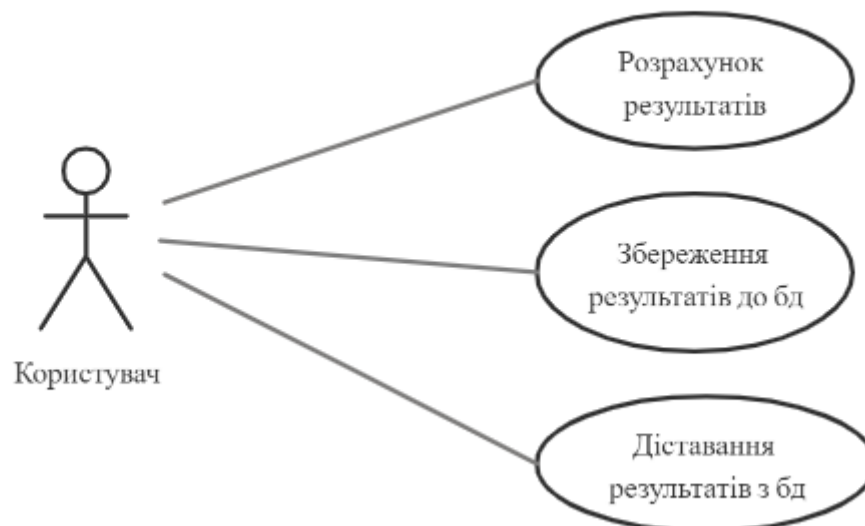


Рисунок 4.1 — Діаграма прецедентів

Під час авторизації, користувач надсилає запит на сервер. Після цього він матиме змогу працювати з підсистемою. Для запуску процесу розрахунку даних користувач підсистеми вводить необхідні вхідні данні.

Сама структура проекту виглядає як зображено на рис 4.2

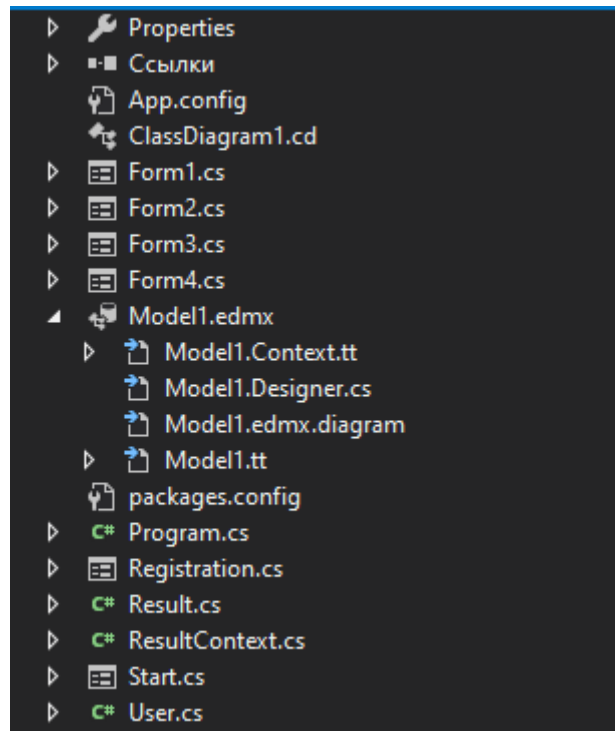


Рисунок 4.2 — Структура проекту

Частина на якій працює клієнт написана на C# з використанням технології Windows Forms. Windows Forms — інтерфейс програмування додатків (API), відповідальний за графічний інтерфейс користувача і є частиною Microsoft .NET Framework. Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за допомогою створення обгортки для Win32 API в керованому коді[7]. Завдяки Windows Forms можна створювати безліч так названих форм які зручні для кінцевого користувача та інтуїтивно зрозумілі в користуванні. В цілому в проекті є 5 форм які використовуються та 3 активні класи. Одна з головних форм показана на рисунку 4.3

The screenshot shows a window titled 'TEF'. It contains a table with 10 columns: №, Vcm, V3, V31, R1\*10³, t, TauW, Re, and G3 (orig Re). The first row has an asterisk in the '№' column. Below the table is a large gray rectangular area. At the bottom of the window are four buttons: 'Input Data...', 'Start', 'Save', and 'Load...'.

№	Vcm	V3	V31	R1*10³	t	TauW	Re	G3 (orig Re)
*								

Рисунок 4.3 — Головна форма

За кожною формою закріплений і свій клас, тобто фактично в формі більше класів проте будемо вважати їх як форми. Почнемо з першої форми Start.cs. Вікно форми показано на рисунку 4.4

The screenshot shows a window titled 'Start'. It contains a 'Login' label above a text box with 'Admin' entered. Below that is a 'Password' label above a text box with '\*\*\*\*\*' entered. At the bottom are two buttons: 'Ok' and 'Registration'.

Рисунок 4.4 — Форма входу

На цій формі декілька полів для вводу та дві кнопки. Більш цікавий сам код форми. На рисунку 4.5 показаний конструктор класу.

```

public partial class Start : Form
{
    ResultContext db;
    public Start()
    {
        InitializeComponent();
        db = new ResultContext();

        db.Users.Load();
    }

    private void Button1_Click(object sender, EventArgs e)
    {
        bool fl = false;
        var listUsers = db.Users.ToList();

        foreach (var user in listUsers)
        {
            if (user.Login.Equals(LogTextBox.Text))
            {
                if (user.Password.Equals(PassTextBox.Text))
                {

```

Рисунок 4.5 — Конструктор класу Start.cs

Тут в конструкторі окрім ініціалізації компонентів самих Windows Forms є так званий ResultContext він відповідає за доступ застосунка до бази даних, але більш детально про нього пізніше. Завантажуємо всіх юзерів та при натисненні клавіші вхід перевіряємо правильність даних вже в наступному методі.

```

private void Button1_Click(object sender, EventArgs e)
{
    bool fl = false;
    var listUsers = db.Users.ToList();

    foreach (var user in listUsers)
    {
        if (user.Login.Equals(LogTextBox.Text))
        {
            if (user.Password.Equals(PassTextBox.Text))
            {

                var frm = new Form1();
                this.Hide();
                frm.label1.Text = user.Login;
                fl = true;
                frm.ShowDialog();
                this.Close();
                break;
            }
        }
    }
}

if (fl != true)
{
    MessageBox.Show("Wrong login or password", "Something went wrong", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

```

Рисунок 4.6 —Метод входу Start.cs

При введенні даних користувача відбувається перевірка на правильність логіну та паролю, якщо все правильно то програма завантажує головне вікно. Якщо ж ні то видає помилку.

При натисненні іншої кнопки а саме реєстрації програма завантажить нову форму метод запуску показано на рисунку 4.7

```
private void Button2_Click(object sender, EventArgs e)
{
    while (true)
    {
        var Regist = new Registration();
        DialogResult result = Regist.ShowDialog(this);

        if (result == DialogResult.Cancel)
            return;
        if (result == DialogResult.OK)
        {
            var User = new User() { Login = Regist.LogTextBox.Text, Password = Regist.PassTextBox.Text, Name = Regist.Na
            var listUsers = db.Users.ToList();
            foreach (var user in listUsers)
            {
                if (user.Login.Equals(User.Login))
                {
                    MessageBox.Show("Login already in use", "Something went wrong", MessageBoxButtons.OK, MessageBoxIcon
                    continue;
                }
            }
            db.Users.Add(User);
            db.SaveChanges();
            break;
        }
    }
}
```

Рисунок 4.7 —Метод реєстрації Start.cs

Перейдемо до головної форми. Як вже було сказано в ній відбуваються всі головні розрахунки.



```

using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        ResultContext db;
        public Form1()
        {
            InitializeComponent();

            db = new ResultContext();
            db.Results.Load();
        }

        static List<Result> ListResults = new List<Result>();

        static double Pcm = 1;
        static double Votn = 1;
    }
}

```

Рисунок 4.8 — Конструктор класу Form1.cs

Тут з вже звична нам ініціалізація та завантаження контексту з бази даних. Тільки цього разу ми завантажуюмо саму таблицю результатів а не юзерів.

```

        sum += 2 * func(x);
        x += h;
    }
    sum = (h / 3) * (sum + func(a) - func(b));

    return sum;
}

static void V3()
{
    v3 = 2 / (Math.Pow(R, 2) - Math.Pow(R1, 2)) * Integrate(x => v3r * (x) * x, R1, R);
    Console.WriteLine("V3=" + v3);
}

static void tauV()
{
    var g = 9.8;

    tauv = ((g * fi * R * (p3 - p1)) / 2) - (R1 / R) * (((g * fi * R1) / 2) * (p3 - p1) + t);
}

```

Рисунок 4.9 — методи розрахунку класу Form1.cs

На рисунку 4.9 наведено декілька методів розрахунків, кожен метод відповідає формулі, також в методі V3() використовується інтеграл, його програмна реалізація наведена на рисунку 4.10

```
public static double Integrate(Func<double, double> func, double a, double b)
{
    double sum = 0;
    double n = 10;
    while (Math.Abs((b - a) / n) > 0.01)
    {
        n++;
    }

    double h = (b - a) / n;

    double x = a + h;
    while (x < b)
    {
        sum += 4 * func(x);
        x += h;
        sum += 2 * func(x);
        x += h;
    }
    sum = (h / 3) * (sum + func(a) - func(b));

    return sum;
}
```

Рисунок 4.10 —методи інтегралу класу Form1.cs

При натисненні кнопки Start запускається метод який в свою чергу запускає інші методи і розраховує всі формули які потрібні для результату. Метод Start представлено на рисунку 4.11

```
{
    tau();
    V3r();
    V3();
    G3();

    V31();
    tauV();
    string[] row1 = new string[] { "#" + "", Vcm + "", v3 + "", v31 + "", R1 * 0.001 + "", t + "", tauV + "", re * 0.0001 + "" };
    dataGridView1.Rows.Add(row1);

    var result = new Result() { R1 = R1, Re = re * 0.0001, t = t, V3 = v3, V31 = v31, Vcm = Vcm, };
    ListResults.Add(result);

    Vcm += +0.4;

    Console.ReadLine();
}
```

Рисунок 4.11 —метод Start класу Form1.cs

Також тут використовується `datagridview` можна було виконати прив'язку даних, але був вибраний варіант з заповненням результатів через масив `string`.

Метод збереження а саме `Save` вибирає поточний список результатів з яким працює користувач та зберігає його до бази даних. Завдяки `entity framework` дані зберігаються у вигляді сутностей тому не має потреби писати складні `sql` запити. Як ідентифактор результатів було вибрано дату, тобто коли користувач захоче знайти результат який він зберіг, йому потрібно просто вказати ту дату коли він зберіг ці результати зі списку, і вони будуть відтворені в `datagridview` на головній формі. Метод `Save` представлений на рисунку 4.12

```
private void Save(object sender, EventArgs e)
{
    var now = DateTime.Now.ToString();
    var now2 = DateTime.Now;
    string v = now2.Day + "" + now2.Month + "" + now2.Year + "" + now2.Hour + "" + now2.Minute + "" + now2.Second;

    Console.WriteLine(now);
    Console.WriteLine(v);
    foreach (var tmp in ListResults)
    {
        tmp.Number = Convert.ToInt64(v);
        tmp.User = db.Users.Where(p => p.Login == label1.Text).FirstOrDefault();
    }
    db.Results.AddRange(ListResults);

    db.SaveChanges();
}
```

Рисунок 4.12 —метод `Save` класу `Form1.cs`

Метод `Load` відкриває нову форму та пропонує користувачу завантажити один з його результатів які він зберіг раніше. Спочатку йде завантаження всіх результатів які належать користувачу і виводиться їх ключ тобто дата як ідентифікатор. При виборі користувачем одного з результатів він виводиться на головну форму. Метод `Load` представлений на рисунку 4.13.

```
private void Load(object sender, EventArgs e)
{
    db.Results.Load();
    var LoadForm = new Form3();

    var groups = from p in db.Results
                  where p.User.Login == label1.Text
                  group p by p.Number;
    foreach (var g in groups)
    {
        LoadForm.listBox1.Items.Add(g.Key);
    }
    LoadForm.label1.Text = label1.Text;

    DialogResult result = LoadForm.ShowDialog(this);
    LoadForm.listBox1.SelectedItem = 0;
    if (result == DialogResult.Cancel)
        return;
    long s = Convert.ToInt64(LoadForm.listBox1.SelectedItem);

    var groups1 = db.Results.Where(p => p.Number == s);

    dataGridView1.Rows.Clear();|
}
```

Рисунок 4.13 —метод Load класу Form1.cs

А відображення результатів відбувається вже знайомим нам способом на рисунку 4.14

```
foreach (var p in groups1)
{
    string[] row1 = new string[] { p.Id + "", p.Vcm + "", p.V3 + "", p.V31 + "", p.R1 + "", p.t + "", tauv + " --"
    dataGridView1.Rows.Add(row1);
}
button2.Enabled = false;
```

Рисунок 4.14 —відображення результатів Form1.cs

Інші форми не мають в своїх класах якусь логіку або складні методи. Вся логіка відбувається в головній формі а форму служать як зворотній зв'язок від користувача.

Клас User.cs представлений на рисунку 4.15

```

Ссылка: 4
class User
{
    Ссылка: 0
    public int Id { get; set; }
    Ссылка: 1
    public string Name { get; set; }
    Ссылка: 8
    public string Login { get; set; }
    Ссылка: 2
    public string Password { get; set; }

    Ссылка: 1
    public ICollection<Result> Results { get; set; }

    Ссылка: 1
    public User()
    {
        Results = new List<Result>();
    }
}

```

Рисунок 4.15 — Клас User.cs

Аналогічний за структурою і клас Result.cs представлений на рисунку 4.16

```

class Result
{
    Ссылка: 1
    public int Id { get; set; }
    Ссылка: 3
    public long Number { get; set; }

    Ссылка: 2
    public User User { get; set; }
    Ссылка: 2
    public double Vcm { get; set; }
    Ссылка: 2
    public double V3 { get; set; }
    Ссылка: 2
    public double V31 { get; set; }
    Ссылка: 2
    public double R1 { get; set; }
    Ссылка: 2
    public double t.1 { get; set; }
    Ссылка: 0
    public double t.w { get; set; }
    Ссылка: 2
    public double Re { get; set; }
    Ссылка: 0
    public double Re3 { get; set; }
}

```

Рисунок 4.16 — Клас Result.cs

Самим головним в роботі Entity Framework прийнято вважати класс ResultContext.cs представлений на рисунку 4.16

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WindowsFormsApp1
{
    Ссылка: 5
    class ResultContext: DbContext
    {
        Ссылка: 2
        public ResultContext() : base("DefaultConnection") { }
        Ссылка: 4
        public DbSet<Result> Results { get; set; }
        Ссылка: 5
        public DbSet<User> Users { get; set; }
    }
}
```

Рисунок 4.16 — Клас Result.cs

На перший погляд в класі не дуже багато коду, проте цей клас успадковується від іншого а саме DbContext в якому і описана вся логіка який і являє собою основу Entity framework також в цьому класі є так звані DbSet з юзерами та результатами які і зберігають інформацію.

## Висновок до розділу 4

У даному розділі було наведено опис реалізації програмного продукту. Була представлена діаграма прецедентів та описані основні сценарії використання продукту, а також їх програмна реалізація. Було описано створення DbContext, а саме створення зв'язку з сервером бази даних за допомогою Entity Framework

Для реалізування клієнтської частини було використано бібліотеку Windows Forms. З її допомогою було реалізовано гнучкий і зручний інтерфейс,

можливість робити запити до серверу бази даних за допомогою Entity framework і надано інтерфейсу інтерактивності.

## 5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Для забезпечення безвідмовної роботи програмної підсистеми чисельного моделювання треба дотримуватися основних вимог при інсталяції та рекомендацій щодо її використання.

### 5.1. Системні вимоги та запуск

Для запуску розробленої програмної системи персональний комп'ютер повинен мати двох-ядерний процесор Intel ® чи AMD з тактовою частотою не нижче 2.0 GHz, відеокарта серії NVIDIA GeForce 480 GT або ж вбудоване у процесор графічне ядро, 4 ГБ оперативної пам'яті, на комп'ютері повинна бути встановлена операційна система Windows 7, Windows 8, Windows 10, а також .Net framework 4.7.0. Необхідно також встановлений SQL Server. Також на жорсткому диску повинно бути не менше 350 Мб вільного місця.

Для запуску системи необхідно запустити виконавчий файл TEF.exe (рисунок 5.1)

Имя	Дата изменения	Тип	Размер
EntityFramework.dll	14.09.2019 5:01	Расширение при...	4 872 КБ
EntityFramework.SqlServer.dll	14.09.2019 5:01	Расширение при...	578 КБ
EntityFramework.SqlServer.xml	14.09.2019 5:01	Документ XML	160 КБ
EntityFramework.xml	14.09.2019 5:01	Документ XML	3 650 КБ
<b>TEF.exe</b>	<b>27.11.2019 10:04</b>	<b>Приложение</b>	<b>41 КБ</b>
WindowsFormsApp1.exe.config	20.11.2019 5:52	XML Configuratio...	1 КБ
WindowsFormsApp1.pdb	27.11.2019 10:04	Program Debug D...	86 КБ

Рисунок 5.1 — Запуск виконавчого файлу



Після цього потрібно або увійти під своїм логіном або ж зареєструватись. Вікно входу в програму показана на рисунку 5.2

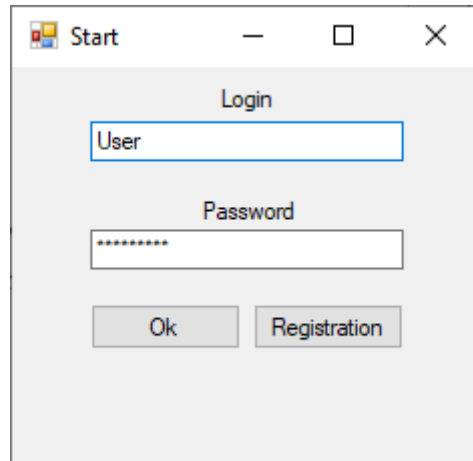


Рисунок 5.2 — Вікно входу в програму

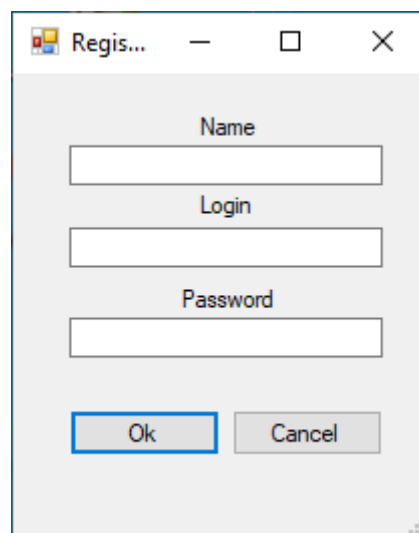
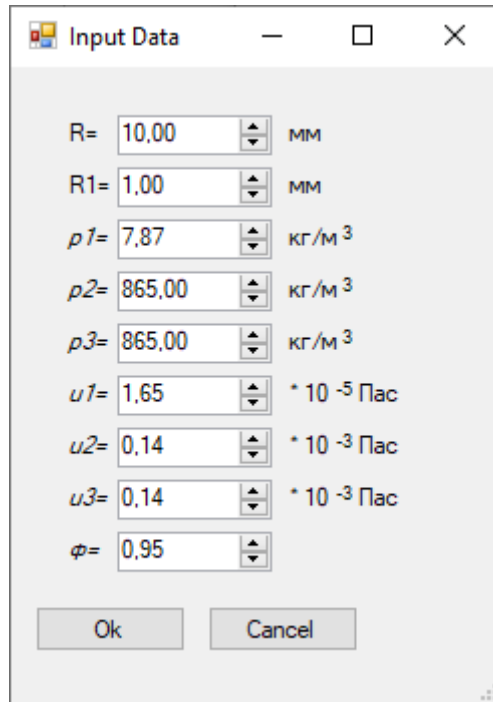


Рисунок 5.3 — Вікно реєстрації

Далі потрібно ввести вхідні данні у вікно під назвою Input Data. Вікно для вводу даних показано на рисунку 5.4.



Input Data

R= 10,00 мм

R1= 1,00 мм

$\rho_1$ = 7,87 кг/м<sup>3</sup>

$\rho_2$ = 865,00 кг/м<sup>3</sup>

$\rho_3$ = 865,00 кг/м<sup>3</sup>

$\nu_1$ = 1,65 \* 10<sup>-5</sup> Пас

$\nu_2$ = 0,14 \* 10<sup>-3</sup> Пас

$\nu_3$ = 0,14 \* 10<sup>-3</sup> Пас

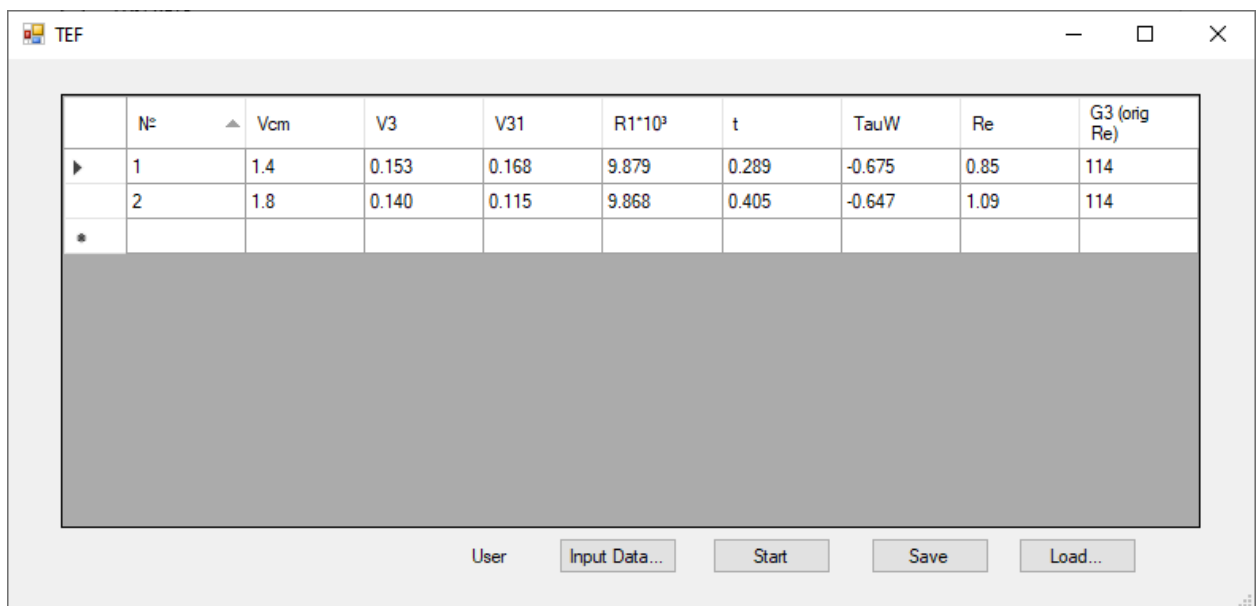
$\phi$ = 0,95

Ok Cancel

Рисунок 5.4 — Вікно для вводу вхідних даних

Після цього програма проведе обчислення на основі вхідних даних та виведе їх на вікно результатів.

Вікно для виводу результатів показано на рисунку 5.5



TEF

	N°	Vcm	V3	V31	R1*10 <sup>3</sup>	t	TauW	Re	G3 (orig Re)
▶	1	1.4	0.153	0.168	9.879	0.289	-0.675	0.85	114
	2	1.8	0.140	0.115	9.868	0.405	-0.647	1.09	114
*									

User Input Data... Start Save Load...

Рисунок 5.4 — Вікно для виводу результатів

Користувач може зберегти данні натиснувши на кнопку Save тоді результати зберезуться в базі даних. Щоб переглянути результати обчислень користувача які він виконував раніше, потрібно натиснути кнопку Load... та у новому вікні з переліку всіх результатів вибрати той який його цікавить. Вікно завантаження результатів показано на рисунку 5.5

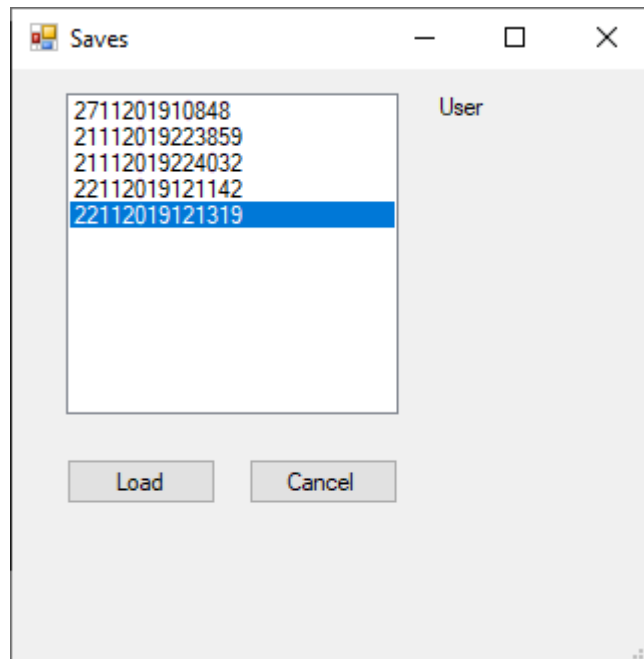


Рисунок 5.5 — Вікно завантаження результатів

Після вибору результата він відобразиться в вікні для виводу результатів.

## Висновок до розділу 5

У даному розділі були описані основні вимоги до інсталяції та рекомендації щодо використання програмного продукту.

Було описано можливі сценарії взаємодії користувача з додатком також були описані та наглядно продемонстровані основні можливості програмного

продукту, такі як запуск системи, роботу з головним вікном та вікнами для завантаження результатів.

## 6. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

Стартап як форма малого ризикового підприємництва впродовж останніх десяти років набула широкого розповсюдження у світі через зниження бар’єрів входу в ринок (з появою Інтернету як інструменту комунікацій та збуту стало значно простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

### 6.1. Опис ідеї проекту

Основною метою даної дисертації є розробка Підсистема чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів. Опис представлено на таблиці 6.1:

Таблиця 6.1 — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Підсистема чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів	1. Розрахунок результатів критичних режимів на основі формул моделі кільцевої течії	1. Розрахунок результатів на критичних режимів роботи теплообмінних апаратів
		2. Збереження розрахункових результатів до бази даних

Було визначено технологічну здійсненність ідеї проекту у таблиці 6.2:

Таблиця 6.2 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технологія її реалізації	Наявність технологій	Доступність технологій
1	Обрати платформу для вирішення задачі	Клієнт-серверний додаток	Веб-сервери на, Java, C#, Node.js і тд.	Доступні
		Мобільний додаток	Додатки на Java, Xamarin, Kotlin, Swift.	Доступні
		Додаток для ПК	Додатки на Java, C#, Node.js і тд..	Доступні
2	Обрати бібліотеку для роботи з базою даних	.Net модуль що робить віддалені запити	Entity Framework	Доступна безкоштовно
3	Обрати сервер для бази даних	Сервер бази даних який зберігає інформацію	Sql Server від Microsoft	Доступна безкоштовно
4	Розробити систему для розрахунків	сервер на .Net	Entity framework	Доступна безкоштовно
		додаток на C#	.net framework	Безкоштовна Community версія

Проведена робота показує, що для ідеї проекту є можливою. Модуль розрахунків за формулами розміщується окремо від основних модулів системи для того, щоб у подальшому можна було використати будь-яке джерело з відкритих джерел Все буде залежати від програмної реалізації, але до системи можна буде підключити будь-яку електронну бібліотеку, або ж пошукову

систему. Було обрано реалізацію ідеї у вигляді клієнт-серверного додатку з використанням С# модулів. Такий вибір надає можливість використовувати велику кількість аналітичних, потужних модулів для обробки великої кількості інформації. Це дозволило у досить короткий термін розробити першу доступну програмну реалізацію.

Ця система, власне, і дозволила провести аналіз точності використовуваного алгоритму. Після цього і було прийнято рішення про модифікацію використовуваного алгоритму.

Для модифікації алгоритму було знову ж таки використано бібліотеку sklearn, так як вона надавала такі можливості. Так як використовувався векторний метод порівняння, було вирішено модифікувати його додаванням перевірки

У таблиці 6.3 було визначено ринкові можливості, які можна використати під час ринкового впровадження проекту, та ринкові загрози, які можуть перешкодити реалізації проекту. Все це дозволить спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Таблиця 6.3 — Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	10
2	Загальний обсяг продаж, грн/ум.од	20
3	Динаміка ринку (якісна оцінка)	зростає

4	Наявність обмежень для входу	немає
5	Специфічні вимоги до стандартизації та сертифікації	Розмістити застосунок та налаштувати сервер.
6	Середня норма рентабельності в галузі (або по ринку), %	0.5

Проведений аналіз показав, що проект не є дуже прибутковим, тому не є дуже привабливим для вкладень. Проте можливо є сенс вкладати гроші у розробку таких швидких і точних алгоритмів підрахунку результатів. Адже основною ідеєю даного проекту є перевірка життєздатності розробленого модулю а не самої системи. Проведена робота дозволяє зробити висновок, що алгоритм є життєздатним і може бути інтегрованим у більш складні системи, які будуть ділити логіку системи на окремі модулі або фрагменти і використовувати ці модулі незалежно від системи в якій вони використовуються.

Слід відзначити легкість впровадження розробленої системи для роботи у навантаженому режимі. Потрібно розмістити додаток та налаштувати сервер.

Далі було визначено основні групи клієнтів, як показано на таблиці 6.4:



Таблиця 6.4 — Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Необхідно мати можливість перевірити робити розрахунки на основі вхідних даних	Цільовою аудиторією проекту є наукові працівники, наукові структури, інженерно-технічних працівники	Відсутня	Єдиною вимогою до користувачів до товару є або реєстрування у доступному додатку, або ж встановлення системи на власних серверах

Клієнтами даної системи можуть бути різні люди, які приймають участь у публікації статей, організації роботи контактних теплообмінних апаратів, а також інженерно-технічні працівники.

Для того щоб користуватися даною системою, користувачу необхідно буде зареєструватися у системі. Після цього він зможе заходити до свого профілю і виконувати. Також користувач зможе зберігати результати на сервер та діставати данні звідти

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища. Фактори загроз представлено у таблиці 6.5:

Таблиця 6.5 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Устарювання алгоритму	З часом алгоритм може втратити свою ефективність у порівнянні з іншими присутніми на ринку	Вкласти часові ресурси у розроблення більш ефективного алгоритму
2	Недоступність серверу	Може статися що застосунок може бути недоступним внаслідок проблем провайдера чи перевантаження	Збільшити ресурси для серверів, додати захист від вірусних атак.

Також були проаналізовані основні фактори можливостей у таблиці 6.6:

Таблиця 6.6 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Додання можливості порівнювати результати	Якщо система стане успішно порівнювати результати між результатами користувача або інших користувачів це дасть змогу робити обширні висновки	Розробити алгоритм порівняння результатів між користувачами
2	Додання можливості ділитись вже готовими або популярними результатами	Було б дуже зручно ділитися результатами які дуже популярні в розрахунках, щоб не виконувати розрахунки по декілька разів.	Розробити модуль який надаватиме змогу ділитися результатами

. Проведений аналіз дозволяє зробити висновок, що існує немало шляхів для вдосконалення поточної системи. Існують фактори загроз до яких необхідно бути готовим при виході на стартап ринок. Також існують можливості, які дозволять зробити декілька впевнених кроків у засвоєнні ринку.

Також було проведено більш детальний аналіз умов конкуренції галузі в таблиці 6.7:

Таблиця 6.7 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Sifer, KensPaul, Rebevgotus	Входження у ринок обмежене лише часом налаштування робочого середовища системи	Провайдерам робочого середовища може бути SCAR або ж CPS	Споживачі будуть раді виходу на ринок дешевої системи перевірки на плагіат	Замінниками можуть бути ті ж самі конкуренти
Висновки:	Конкуренція є дуже високою, але у деяких випадках переваги полягають у швидкодії системи або ж низька ціна входу, або ж більша доступність	Вихід на ринок не є дуже складною задачею, необхідно налаштувати робоче середовище, а це може бути виконано за тиждень	Так, провайдери серверів звичайно диктують свої умови, наприклад ціна забезпечення серверів. Проте ці провайдери відомі як дуже надійні і вони забезпечують інструменти для забезпечення високої доступності системи і її потужності	Так, звичайно і клієнти диктують умови. Наприклад, якщо випустити систему, яка робить те, що нікому не потрібно, то на цій системі і не вийде нічого заробити. Тому система повинна реалізувати функціонал, який є необхідним для користувачів і за який вони готові платити	Товарі замінників не передбачено

Проведений аналіз дозволяє зробити висновок, що на ринку представлена не слабка конкуренція. Проте якщо всі аспекти системи будуть виконані так як заплановано, то це дасть системі значну перевагу перед конкурентами наприклад, якщо буде гарно реалізовано модуль доступу до бази даних то він буде легко замінитися іншими системами управління баз даних. Така модифікація дозволить розробити такі собі продукти, які будуть працювати на різних нішах ринку і розповсюджувати вплив самої системи на всьому ринку.

На основі аналізу конкуренції, а також із урахуванням характеристик ідеї проекту, вимог споживачів до товару та факторів маркетингового середовища у таблиці 6.8 було визначено перелік факторів конкурентоспроможності.

Таблиця 6.8 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Застосунок використовує формули моделі кільцевої течії	Застосунок використовує формули моделі кільцевої течії, що дозволяє значно покращити точність розрахунків.
2	Окремий модуль для роботи с базою даних	Використовується окремий модуль для роботи с базою даних, що робить можливим у майбутньому використання будь якої системи яка відповідає тим же властивостям і налаштування системи під різні середовища.
3	Клієнт-серверна архітектура	Використовується підхід клієнт-серверної архітектури що дозволяє окремо підвищувати потужність серверу, незалежно від клієнтської частини і економити ресурси

У таблиці 6.9 було проведено аналіз сильних і слабких сторін стартап-проекту:

Таблиця 6.9 — Порівняльний аналіз сильних та слабких сторін “Система аналізу збіжності текстової інформації для оцінки плагіату”

№ п/п	Фактор конкурентноспроможності	Бали	Рейтинг товарів конкурентів у порівнянні з Sifer						
			-3	-2	-1	0	1	2	3
1	Формули моделі кільцевої течії	10				+			
2	Окремий модуль для роботи с базою даних	16							+
3	Клієнт-серверна архітектура	12						+	

Для розроблення ринкової стратегії першим кроком є визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів показано у таблиці 6.10.

Таблиця 6.10 — Вибір цільових груп потенційних користувачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Науковці	Дуже необхідно мати інструмент для перевірки тих чи інших даних для своїх робіт тому така система вирішить усі питання з правильності	Більшість тих хто ще не є клієнтом конкуренту	Конкуренція не є дуже інтенсивною	Немає перешкод

		підрахунків			
2	Інженерно-технічні працівники	Можна перевіряти правильність результатів при експлуатації	Не дуже високий, адже зазвичай спеціаліст може легко відрізнити оригінальну думку від плагіату	Конкуренція не є дуже інтенсивною	Немає перешкод
3	Розробники тепло-масообмінних апаратів	Дуже необхідно мати інструмент тих чи інших даних для своїх розробок в плані правильності експлуатації різних модулів	Більшість тих хто ще не є клієнтом конкуренту	Конкуренція не є дуже інтенсивною	Немає перешкод
Які цільові групи обрано: науковці, інженерно-технічні працівники, розробники тепло-масообмінних апаратів					

Далі було сформовано базову стратегію розвитку, як показано у таблиці 6.11:

Таблиця 6.11 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект “першопрохідцем” на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів,	Чи буде компанія копіювати основні характеристики товару конкуренту, і які?	Стратегія конкурентної поведінки
-------	--	--	---	----------------------------------

	Ні	Метою є популяризація систем чисельного моделювання режимів роботи контактних тепло-масообмінних апаратів. Тобто планується пошук нових користувачів із тих хто раніше ніколи не використовував таке ПЗ. Проте якістю і меншою ціною можна буде зазіхнути на користувачів конкурентів	Можливо копіювати якісь елементи інтерфейсу користувача, що покращує задоволення від користування системою	Стратегія виклику лідера
--	----	---	--	--------------------------

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також в залежності від обраної стратегії розвитку та стратегії конкурентної поведінки була розроблена стратегія позиціонування, що полягає у формуванні ринкової позиції, за яким споживачі мають ідентифікувати торгівельний проект.

Результатом проведення даного аналізу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначатиме напрями роботи стартап-компанії на ринку.

Далі необхідно було сформулювати маркетингову концепцію потенційного товару. Для цього було створено таблицю 6.12:

Таблиця 6.12 — Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
-------	---------	----------------------------	-------------------------------------

1	Розрахунок режимів роботи контактних тепло-масообмінних апаратів	Розрахунок результатів за формулами моделі кільцевої течії	Перевагою є використання алгоритму за допомогою якого забезпечується швидкодія системи
2	Додання можливості порівнювати результати	Може надавати таку можливість	Перевагою є відсутність таких характеристик у конкурентів

### **Висновок до розділу 6**

Проведений аналіз дозволяє зробити висновок, що є можливість ринкової комерціалізації проекту. На ринку наявний попит на дані послуги, конкуренція не є надто щільною та існує функціонал, якого потребує користувач, але поки ніхто не може його надати. Подальша імплементація проекту є доцільною.



## ВИСНОВКИ

Тема розробки підсистема чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів є досить цікавою та представляє широке поле для подальших досліджень.

Метою даної роботи було використання архітектури клієнт-сервер для створення програмного продукту, який представляє собою систему чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів. Розрахунок інформації відбувається за формулами моделі кільцевої течії.

Інструментарій для виконання роботи було обрано з огляду на його функціональність та відповідність нашим вимогам для даної роботи. Entity framework дозволив дуже зручно працювати з базою даних Sql Server на рівні сутностей. Система написана мовою C# частина клієнта реалізована за допомогою Windows Forms. Середовище програмування Microsoft Visual Studio дозволило писати код програми, і зручно проводити процес відлагодження та пошуку помилок у програмі.

Проведений аналіз показав, що за допомогою програмного середовища C# та всіх перерахованих технологій (Entity Framework, Windows Forms, NuGet) можна розробити підсистема чисельного моделювання критичних режимів роботи контактних тепло-масообмінних апаратів. У якості алгоритму було обрано формули моделі кільцевої течії.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Client-Server Interaction over the Internet – Режим доступу: <http://www.icodeguru.com/dotnet/core.c.sharp.and.dot.net/0131472275/ch16le1sec1.html>. – Дата доступу: 18.07.2019.
2. Gerhard M. Securing and Optimizing Linux: RedHat Edition version 1.3, RedHat, 2000. 486 с.
3. Client–server model [Електронний ресурс] – Режим доступу: [https://en.wikipedia.org/wiki/Client%E2%80%93server\\_model](https://en.wikipedia.org/wiki/Client%E2%80%93server_model)
4. В. Олифер, Н. Олифер "Компьютерные сети. Принципы, технологии, протоколы" (2016)
5. Ключев Н.И., Соловьева Е.А. Математические модели двухфазных течений: Учебное пособие. - Самара: Изд-во «Самарский университет», 2010. – 51с.
6. Ключев Н.И., Соловьева Е.А. Квазигомогенная модель дисперсно- пленоч-ного течения двухфазной смеси // ИВУЗ Авиационная техника. 2005, №4. С. 35-38.
7. Джеффри Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# (2016) 896с.
8. Microsoft Visual Studio [Електронний ресурс] – Режим доступу: [https://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio)
9. Entity Framework [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp/entityframework/1.1.php>
10. Сара Морган, Тобиаш Тернстрем Проектирование и оптимизация доступа к базам данных Microsoft SQL Server 2005 480с.
11. NuGet [Електронний ресурс] – Режим доступу: <https://en.wikipedia.org/wiki/NuGet>

12. C Sharp [Электронный ресурс] – Режим доступа:  
[https://ru.wikipedia.org/wiki/C\\_Sharp](https://ru.wikipedia.org/wiki/C_Sharp)